

L2 NS Distributed Systems

Summative Assignment – Just Hungry

You are required to implement a fault-tolerant distributed system, called “Just Hungry”, based on passive replication, supporting online food ordering and delivery. You are expected to use RMI in either Java or Python implementing the system. There are various aspects for constructing such a system, including client and server implementation, operation workflow design, data integrity, message design, etc. Your implementation should comprise at least three back-end servers, one front-end server, supporting a client to retrieve and submit food orders. The coursework allows you to demonstrate your competence in developing distributed systems. Submission deadline is 13th March 2020 (2pm). This assessment contributes 12.5% of the overall COMP2211 module mark.

Requirements and Marking Criteria:

- System Design (10 marks): Using a diagram, illustrate major operation workflow flows among back-end servers, the front-end server and a client, and show how they work together as a distributed system.
- Back-end Server Implementation (40 marks): With passive replication, implement at least three back-end servers to accept, process and make responses to user orders. The implementation should fulfill the location, relocation, replication and failure transparency requirements.
- Front-end Server Implementation (20 marks): Implement a front-end server to provide one system access, preventing a client to have direct access to individual system components and supporting transparency requirements.
- Client Implementation (15 marks): Implement a client program to allow a user to access the required distributed system by making food ordering requests and receiving system responses.
- Use of Web Services (15 marks): Incorporate external Web Services to support delivery address retrieval by postal code. The implementation should treat Web Services as distributed system components and fulfill suitable transparency requirements.

For implementation, you should apply knowledge and methods you have learnt from the lectures and practical sessions. The marking criteria shows the breakdown of how the coursework requirements will be marked. As shown in Table 1, the percentage of marks awarded under each of the criteria will be determined by the level of achievement you have made. Your implementation is expected to focus on the distributed system aspects. You may create a minimal amount of sample data to help you demonstrate the system functionalities and how the system meets the requirements. You may also implement your work using a text-based user interface and a memory-based data structure (e.g. array) to store and update system information. Other irrelevant features including database design and graphical user interface are not required.

Your submission should include the system design diagram, all source codes and resource files, a readme file showing instructions of how to run your system and what external resources you have adopted. You should compress all files into a single zip file and upload it to DUO for submission.

Table 1:

Level of achievement	Percentage of Marks
No implementation	0%
Inadequate or incomplete implementation	0 - 40%
Satisfactory to Good (in terms of correctness and completeness)	40 - 60%
Very Good to Excellent (in terms of completeness and robustness)	60 – 80%
Outstanding to Perfect (in terms of completeness, robustness and complexity)	80 - 100%